

A Study of Metrics of Distance and Correlation Between Ranked Lists for Compositionality Detection

Christina Lioma and Niels Dalum Hansen

Department of Computer Science, University of Copenhagen, Denmark

Abstract

Compositionality in language refers to how much the meaning of some phrase can be decomposed into the meaning of its constituents and the way these constituents are combined. Based on the premise that substitution by synonyms is meaning-preserving, compositionality can be approximated as the semantic similarity between a phrase and a version of that phrase where words have been replaced by their synonyms. Different ways of representing such phrases exist (e.g., vectors [1] or language models [2]), and the choice of representation affects the measurement of semantic similarity.

We propose a new compositionality detection method that represents phrases as ranked lists of term weights. Our method approximates the semantic similarity between two ranked list representations using a range of well-known distance and correlation metrics. In contrast to most state-of-the-art approaches in compositionality detection, our method is completely unsupervised. Experiments with a publicly available dataset of 1048 human-annotated phrases shows that, compared to strong supervised baselines, our approach provides superior measurement of compositionality using any of the distance and correlation metrics considered.

Keywords: Compositionality Detection, Metrics of Distance and Correlation

1. Introduction

Compositionality in natural language describes the extent to which the meaning of a phrase can be decomposed into the meaning of its constituents and the way these constituents are combined. Automatic compositionality detection has long received attention (see, for instance, [1], [3], [4]). One popular compositionality detection technique is to replace the constituents of a phrase by their synonyms (one at a time), obtaining a new “perturbed” phrase, and to measure the semantic distance of the resulting phrase to the original phrase: the smaller the semantic distance, the higher the degree of compositionality in the original phrase. Table 1 illustrates this point.

Different variants of this substitution-based approach to compositionality detection exist (overviewed in Section 2), which differ in their measurement of semantic similarity between the original and the perturbed phrase. One approach is to represent

Email address: {c.lioma,nhansen}@di.ku.dk (Christina Lioma and Niels Dalum Hansen)

Original Phrases	Perturbed Phrases	Semantic Distance	Compositionality
red car	scarlet car, red vehicle	small	high
red tape	scarlet tape, red ribbon	large	low

Table 1: Examples of a very compositional (red car) and a non-compositional (red tape) phrase, and their perturbations.

the original phrase and its perturbed phrases as vectors, whose elements are contextual terms (i.e. frequently co-occurring terms) extracted from their respective distributional semantics in some appropriate corpus. Then, vector distance can be measured using, for instance, the cosine of the angle of the vectors. Another approach is to represent the original and perturbed phrases as language models and approximate their similarity, e.g., using Kullback-Leibler divergence. These are mostly *supervised* approaches that consider the aggregated distance or divergence (over all the elements of the vectors or language models) as an (inverse) approximation of semantic similarity.

We conjecture that it is not necessary to use all the elements of such vectors or language models to compute semantic similarity, but that it suffices to use solely the most semantically salient elements. If this conjecture is true, the potential benefit is that semantic similarity (and any derived analysis thereof) can be done more efficiently as, e.g., the involved computations will be done on lower dimension elements. Motivated by this, we propose the following *alternative approach to compositionality detection*: we represent the original phrase and its perturbed phrases as ranked lists, containing only the term weight (e.g., TF-IDF [5]) of their contextual terms. We rank the elements of each list decreasingly by this term weight. This allows to compute the semantic similarity between the phrase and its perturbed phrases using a range of different metrics designed for ranked list comparison. Different such metrics emphasise different aspects of the ranked lists, such as the weight of the elements in the list, or their position in the ranking, or the similarity between elements. As metrics of distance and correlation between ranked lists tend to be parameter-free, our approach is completely unsupervised.

Our contribution: We propose a novel formulation of compositionality detection as ranked list similarity that is completely unsupervised. We perform an empirical study of how this formulation fares when a number of different distance and correlation metrics for ranked lists are used. The results of the study show that our method performs better than strong, recent, supervised baselines.

2. Related Work on Compositionality Detection

We divide related work on compositionality detection into two broad categories:

- approaches estimating the similarity between a phrase and its components (mostly earlier), and
- approaches estimating the similarity between a phrase and a perturbed version of that phrase where terms have been replaced by their synonyms (more recent).

Baldwin et al. [4] use Latent Semantic Analysis (LSA) to calculate the similarity between a phrase and its components, reasoning that higher similarity indicates higher

compositionality. They extract contextual vectors for the terms in the phrase, and represent them as points in vector space. They measure the similarity between two vectors as the cosine of the angle between them. Katz and Giesbrecht [6] present a similar idea (with a slightly different choice of methods and with German data). Venkatapathy and Joshi [7] also present a similar idea, by extending the LSA context vectors of Baldwin et al. [4] with collocation features (e.g. phrase frequency, point-wise mutual information) extracted from the British National Corpus. More recently, Reddy et al. [8] define a term literality score as the similarity between a phrase and its constituent contextual vectors. They use different vectorial operations to estimate the semantic distance between a phrase and its individual components, from which they deduce compositionality.

Closer to ours is the work of Kiela and Clark [1], who detect non-compositionality based on the earlier hypothesis that the mutual information between the constituents of a non-compositional phrase is significantly different from that of a phrase created by substituting terms in the original phrase by their synonyms [3]. They represent phrases by their context vectors. Using standard vectorial similarity, this model slightly outperforms that of McCarthy et al. [9] and Venkatapathy and Joshi [7]. A recent variation of this idea replaces the context vectors with language models and computes their Kullback-Leibler divergence to approximate their semantic distance [2]. However, the accuracy of this approach has not been evaluated.

Further approaches to compositionality detection also exist. For instance, Cook et al. [10] use syntax to identify non-compositionality in verb-noun phrases. They reason that compositional expressions are less syntactically restricted than non-compositional ones, as the latter tend to occur in a small number of fixed syntactic patterns. Along a similar line, McCarthy et al. [9] consider the selectional preferences of verb-object phrases.

Lastly, compositionality detection has also been studied using representation learning of word embeddings. Socher et al. [11] present a recursive neural network (RNN) model that learns compositional vector representations for phrases and sentences of arbitrary syntactic type and length. They use a parse tree structure and assign a vector and a matrix to every node: the vector captures the meaning of the constituent, while the matrix captures how it changes the meaning of neighboring words or phrases. Mikolov et al. [12] extend a word-based skip-gram model that learns non-compositional phrases by being trained on phrases (treated as individual tokens) as opposed to individual words. The training phrases are extracted from a corpus using a threshold on the ratio of their bigram over (product of) unigram counts. Along a similar line, Salehi et al. [13] use the word embeddings of Mikolov et al. [14] with several vectorial composition functions to detect non-compositionality. Yazdani et al. [15] also learn semantic composition and detect non-compositional phrases as those that stand out as outliers in the process. They examine various composition functions of different levels of complexity and conclude that complex functions such as polynomial projection and neural networks can model semantic composition more effectively than the commonly used additive and multiplicative functions.

To our knowledge, no prior work on compositionality detection has used ranked list distance or correlation.

3. Ranked lists for compositionality detection

3.1. Problem formulation

The starting point of our method is the substitution-based approach to non-compositionality detection of Kiela and Clark [1]. Given a phrase, each term is replaced by a synonym (one at a time) producing perturbations of the original phrase by substituting synonyms. The semantic similarity between the original phrase and its perturbations is then assumed to be proportional to the degree of compositionality of the original phrase. This idea is in fact a modern implementation of Leibniz’s principle of inter-substitutivity (*salva veritate*) to detect irregular composition of meaning, which posits that terms which can be substituted for one another without altering the truth of any statement are the same (*eadem*) or coincident (*coincidentia*).

3.2. Our compositionality detection method

Kiela and Clark [1] represent the original phrase and its perturbations as vectors (of their contextual semantics) and compute their semantic distance using cosine similarity, while Lioma et al. [2] represent them as language models and approximate their semantic distance using Kullback-Leibler divergence¹. Instead, we represent the original phrase and its perturbations as *ranked lists*, and use their (inverse) distance or correlation to measure compositionality. The elements of each ranked list are term weights (e.g. TF-IDF) of their contextual terms that are typically represented in vectors. The ranking in each list is by descending term weight, and hence more informative terms are represented earlier in the list.

The high-level steps of our method are displayed in Algorithm I (Table 2).

Algorithm *RankListComp*

Input: Phrase p

Input: Corpus C

Output: Compositionality score $comp(p)$

1. Find synonym \hat{t} of each term $t \in p$
2. **for** each t and \hat{t}
3. $\mathcal{C} \leftarrow$ get context terms from C
4. **for** each context term $i \in \mathcal{C}$
5. compute term weight g_i
6. **for** each \hat{t}
7. Perturbed phrase $\hat{p} \ni \{\hat{t}, |p| - 1 \text{ original terms } t_o\}$
8. List $L_p \leftarrow \text{sort } \{g(i) \forall i \in \mathcal{C}^{\forall t \in p}\}$
9. List $L_{\hat{p}} \leftarrow \text{sort } \{g(i) \forall i \in \mathcal{C}^{\hat{t} \wedge \forall t_o}\}$
10. **return** $\frac{1}{|L_{\hat{p}}|} \sum^{|L_{\hat{p}}|} \text{similarity}(L_p, L_{\hat{p}})$

Table 2: Algorithm I.

¹Strictly speaking, Kullback-Leibler divergence is not a distance.

The ranked lists of the original and perturbed phrase appear in lines 8 – 9 of the above algorithm, respectively. Central in the computation of compositionality is the choice of similarity metric (line 10). A number of different ways to compute this similarity (as distance or correlation) between two ranked lists exist in the literature (for the problem of comparing ranked lists in general, and not for our specific problem formulation). Some metrics consider the aggregate distance over the whole list, whereas other metrics emphasise specific list characteristics, such as the weight, position, or similarity between elements. Next, we present an overview of these metrics.

3.3. Metrics of distance and correlation between ranked lists

Let m and n be positive integers. We consider lists $R_1 = [v_1, \dots, v_m]$ and $R_2 = [w_1, \dots, w_n]$ where $v_1, \dots, v_m, w_1, \dots, w_n \in \mathbb{R}_+ \cup \{0\}$ (i.e., each element in each list is a non-negative real number). We assume that $n \geq m$ (i.e., R_2 is possibly longer than R_1) and $v_1 \geq v_2 \geq \dots \geq v_m$ and $w_1 \geq w_2 \geq \dots \geq w_n$ (i.e., both lists are ordered non-increasingly).

In our problem formulation, R_1 and R_2 represent *ranked lists* of term weights. In principle, the lists may be of unequal length, depending on the implementation of extracting term weights and/or the corpus statistics from which we extract contextual terms. Note also that there is no a priori relationship between the elements of R_1 and R_2 . Indeed in extreme cases we may have $\{v_1, \dots, v_m\} \cap \{w_1, \dots, w_n\} = \emptyset$.

Next, we review metrics on lists that have been, or could be, used to measure some notion of “(dis)similarity” between ranked lists. We partition these metrics into three general classes:

- Class I: Metrics that can natively compute differences between ranked lists of **unequal length**. This is the scarcest class, consisting primarily of modern metrics specifically devised to tackle ranked lists.
- Class II: Metrics that can compute differences between ranked lists of **equal length**. Such metrics can be applied to ranked lists on unequal length only if the *shortest* list (R_1 above) is padded with $n - m$ zeroes at the end, or if the *longest* list (R_2 above) is pruned to the length of R_1 . Both of these options can only work when they do not alter the semantics of the data represented in the list. Most of these Class II metrics are classic metrics on the vector space \mathbb{R}^n .
- Class III: Metrics that can compute differences between ranked lists which constitute **permutations of a set of n elements**. Formally, let $[n] = \{1, \dots, n\}$, and let S_n be the set of permutations on $[n]$; for two elements $\sigma, \pi \in S_n$, a metric d assigns a non-negative real number $d(\sigma, \pi)$ as a “distance”, with $d(\sigma, \pi) = 0$ iff $\sigma = \pi$.

Most of the metrics we have come across are Class III. However, as the ranked lists R_1 and R_2 of term weights that we are interested in may contain different elements and may be of different length, Class III metrics are not directly applicable to our setup of using ranked lists: if R_1 and R_2 were of equal length and contained the same elements, the fact that they are both ordered implies that they would correspond to the exact same permutation of n elements, hence that their distance would be 0 (because they would be

<ul style="list-style-type: none"> • Spearman’s footrule (a.k.a. l_1-distance): total element-wise displacement between two ranked lists. Variations include: weighted, positional, element similarity, and generalised (weighted + positional + element similarity) [16] • Kendall’s τ: total number of pairwise inversions between two ranked lists. Variations include: weighted, positional, element similarity, generalised (weighted + positional + element similarity) [16], version with penalty parameter for swaps early in the permutation [17], and weighted generalisation for ties [18] • Cayley distance: minimal number of transpositions (permutations that swap two adjacent elements) needed to transform element σ to element π [19] • Lee distance: variant of d_{rank} (Eq. 1) where differences $\sigma(i) - \pi(i)$ larger than $\lceil n/2 \rceil$ are instead counted as $n - \sigma(i) - \pi(i)$ [19] • Expected weighted Hoeffding distance: can handle partial or missing rank information [20]

Table 3: Class III metrics (on the set of permutations). The notation of Section 3.3 is used.

identical). Even though Class III metrics cannot be used in our setup, for completeness we briefly outline them in Table 3. Next we focus on Class I and II metrics of distance (Section 3.3.1) and correlation (Section 3.3.2).

We now give a brief overview of the pertinent metrics. We divide the metrics into distances (which, strictly speaking, should satisfy the conditions of being non-negative and symmetric and the shortest possible path between two points) and correlations.

3.3.1. Distance metrics

3.3.1.1 *Rank or l_1 distance (Class III), Minkowski distances (Class II).* Ciobanu and Dinu [21] introduce the *rank distance* d_{rank} (identical to the well-known l_1 distance, a.k.a. Spearman’s footrule). d_{rank} is the sum of absolute rank differences in the lists R_1 and R_2 :

$$d_{\text{rank}}(R_1, R_2) = \sum_{i \in [n]} |\sigma(i) - \pi(i)| \quad (1)$$

While the version of Ciobanu and Dinu [21] is Class III, the l_1 distance can also easily be viewed as being in Class II. Indeed any of the Minkowski distances l_p for $p \geq 1$ induce a distance metric in the usual sense on ranked lists of equal length:

$$l_p(R_1, R_2) = \left(\sum_{i=1}^n |v_i - w_i|^p \right)^{1/p} \quad (2)$$

A similar metric is the *Chebyshev distance*, below.

3.3.1.2 *Chebyshev or l_∞ distance (Class II).* The *Chebyshev* or l_∞ distance is the maximal absolute difference in rank between two equal-sized ranked lists:

$$d_\infty(R_1, R_2) = \max_{i \in [n]} |v_i - w_i| \quad (3)$$

3.3.1.3 CosRank distance (Class III, Class II). Dinu and Ionescu [22] introduce the cosine rank distance d_{CosRank} as the usual cosine distance in an n -dimensional vector space of the vectors $(\sigma(1), \dots, \sigma(n))$ and $(\pi(1), \dots, \pi(n))$:

$$d_{\text{CosRank}}(R_1, R_2) = \frac{\sum_{i \in [n]} \sigma(i) \pi(i)}{\sum_{i \in [n]} i^2} \quad (4)$$

The usual cosine similarity can also be used on the n -dimensional vector space \mathbb{R}^n , hence on equal-sized ranked lists:

$$d_{\text{CosRank}}(R_1, R_2) = \frac{R_1 \cdot R_2}{\sqrt{R_1 \cdot R_1} \sqrt{R_2 \cdot R_2}} \quad (5)$$

where \cdot is the dot product of vectors.

3.3.1.4 Hamming distance (Class II). The *Hamming distance* $d_H(R_1, R_2)$ between equal-sized ranked lists R_1 and R_2 is the number of indices i where $v_i \neq w_i$; using Kronecker's delta δ_{ij} , we can write:

$$d_H(R_1, R_2) = \sum_{i=1}^n (1 - \delta_{v_i w_i}) \quad (6)$$

(equivalently, $d_H(R_1, R_2) = |\{i \in [n] : v_i \neq w_i\}|$).

3.3.1.5 Hausdorff distance (Class I). If a d metric on elements is given, the Hausdorff distance is a particular metric computing distances between certain *sets* of elements. For instance, if one considers the usual Euclidean distance $|\cdot|$ on real numbers, the Hausdorff distance can be used directly on ranked lists of possibly distinct lengths:

$$d_{\text{Haus}}(R_1, R_2) = \max \left\{ \max_{v_i \in R_1} \min_{w_j \in R_2} |v_i - w_j|, \max_{w_j \in R_2} \min_{v_i \in R_1} |v_i - w_j| \right\} \quad (7)$$

The Hausdorff distance is thus the *largest* difference in term weight from a term weight v in R_1 to the term weight in R_2 that is closest to v .

3.3.2. Correlation metrics

Ranked list similarity can also be estimated using *correlation metrics* that generally do not always adhere to all of the classic axioms required by distance metrics in mathematics.

3.3.2.1 Pearson correlation coefficient (Class II). The Pearson (product-moment) correlation coefficient quantifies the linear dependence between two variables. It is the ratio of the covariance of the variables to the product of their standard deviation, easily applicable to two equal-sized ranked lists by using the formula for a sample:

$$d_{\text{Pear}}(R_1, R_2) = \frac{n \sum_{i \in [n]} v_i w_i - \sum_{i \in [n]} v_i \sum_{i \in [n]} w_i}{\sqrt{n \sum_{i \in [n]} v_i^2 - \left(\sum_{i \in [n]} v_i\right)^2} \sqrt{n \sum_{i \in [n]} w_i^2 - \left(\sum_{i \in [n]} w_i\right)^2}} \quad (8)$$

Note that Etesami et al. [23] introduce it as a “Class III” distance, i.e. to compare different rankings of the same group of items. Nevertheless, the usual Pearson correlation can be used directly, as it can be computed for samples, not just for random variables.

3.3.2.2 AP correlation coefficient (Class II). Yilmaz et al. [24] present a correlation metric, the *AP metric*, inspired by Kendall’s τ , that particularly penalizes differences in the top-ranked items. While the metric does not natively apply to lists containing distinct items, it is easy to adapt it to do so.

Let us define $\tau_{ap}(R_1|R_2) = p' - (1 - p')$ where

$$p' = \frac{1}{n-1} \sum_{i=2}^n \frac{C(i)}{i-1} \quad (9)$$

where $C(i)$ is the number of items in R_1 occurring earlier than i whose value is greater than or equal to the value of the i item in R_2 (i.e., in our case $C(i) = \max_{j \in [n], j \leq i} v_j \geq w_i$). $\tau_{ap}(R_2|R_1)$ is defined symmetrically *mutatis mutandis*.

The distance $\text{symm}\tau_{ap}(R_1, R_2)$ is now the following *symmetric* function:

$$\text{symm}\tau_{ap}(R_1, R_2) = \frac{\tau_{ap}(R_1|R_2) + \tau_{ap}(R_2|R_1)}{2} \quad (10)$$

3.3.2.3 Further Class II metrics. Two more correlation metrics that can compute differences between ranked lists of equal length (Class II metrics) are the **Hirschfeld-Gebelein-Rényi maximal correlation** and the **Maximal Rank Correlation** [23]. However, the Hirschfeld-Gebelein-Rényi maximal correlation is a function that, by definition, requires a probability distribution. We do not have such a probability distribution in our setup. The reason we can do without a probability distribution for the Pearson correlation is that it is possible to compute covariance, standard deviation etc. for a sample — but doing so for the Hirschfeld-Gebelein-Rényi maximal correlation and the Maximal Rank Correlation does not seem to make sense without a probability distribution (in particular since the definitions quantify over all functions between two spaces). Also note that no polynomial-time algorithm is known for computing Maximal Rank Correlation, even if an efficiently computable probability mass function is given as input (for instance, by approximating the probability mass functions from relative frequencies).

For compositionality detection, we use all the above distance and correlation metrics (except those discussed in Section 3.2.2.3) to compute the (dis)similarity in line 10 of our algorithm, i.e. the semantic distance between an original phrase and its perturbed version.

Next, we describe in detail the implementation steps of our compositionality detection method.

4. Implementation

We refer the reader to the algorithm of our compositionality detection method (Algorithm I: RankListComp) displayed in Table 2. The remainder explains how each line

in the algorithm is implemented.

4.1. Synonym extraction (Algorithm I, line 1)

The input is some phrase, and the goal is to measure its compositionality. Given an input phrase, for each of its terms, we fetch from WordNet all its hypernyms, and for each hypernym we select all hyponyms. If there are no hypernyms or hyponyms, we fetch all synonyms. We retrieve from the TREC disks 4-5 corpus² all documents that contain any of these hypernyms and hyponyms (and synonyms, if applied), and select the 100 documents where the original term *and* its hypernyms *and* its hyponyms (or synonyms) occur most often. We then select the single hypernym or hyponym (or synonym) that occurs most often in most of these 100 documents. We consider this term as a “near-synonym” to the original input term.

4.2. Contextual terms (Algorithm I, line 3)

For each term (in the original phrase + “near-synonym”) we identify a context window of ± 5 terms around it from TREC disks 4-5 (i.e. the 5 terms preceding the term and 5 terms following it). We neither remove stopwords, nor apply stemming. The above extraction of context windows can give a different number of context terms for different words. Other sizes of context windows can also be used; we use ± 5 terms, following prior work [2].

4.3. Term weights (Algorithm I, line 5)

We compute the TF-IDF score³ of each term t in the context windows as:

$$\sum_{d \in n_t} \frac{f(t, d)}{|d|} \cdot \log_e \frac{N}{n_t} \quad (11)$$

where $f(t, d)$ is the frequency of term t in document d , $|d|$ is the total number of terms in document d , N is the total number of documents in the collection, and n_t is the total number of documents that contain t in the collection. We compute TF-IDF using the corpus statistics of TREC disks 4-5 (unlike [1], who use an adapted version of TF-IDF on the statistics of the context windows). We do this because we aim to compute TF-IDF scores reflecting how generally informative a word is, hence the bigger the corpus used to estimate this, the more representative the resulting scores will be (as long as the corpus has good coverage and representativeness).

²The TREC disks 4-5 corpus is available from: http://www.nist.gov/tac/data/data_desc.html and contains text from a variety of sources: the 103rd Congressional Record, the 1994 Federal Register, the 1992-1994 Financial Times, the 1996 Foreign Broadcast Information Service, and the 1989-1990 Los Angeles Times. We use this corpus because of its coverage and diversity. Any other corpus of reasonable coverage can be used alternatively.

³Any other reliable term weighting score can be used alternatively, for instance any of the simpler [25, 26, 27] or more elaborate [28, 29, 30] formulations in the literature.

4.4. Ranked lists (Algorithm I, lines 8 – 9)

We create ranked lists of TF-IDF scores for each term in the original phrase and its “near-synonyms”. We then combine the ranked lists of TF-IDF scores extracted for each term, into a single ranked list for the whole original phrase (and separately for each perturbed phrase), by simply appending their respective TF-IDF scores to one list and sorting them by their TF-IDF. We do not keep duplicate elements. At this point, we have a single ranked list of TF-IDF scores per phrase (either original or perturbed).

The length of the ranked lists may be different from term to term and may differ across phrases. Indeed the minimum and maximum length of ranked lists we observe per phrase are $\text{min}=200$ (for the perturbed phrase `musculus contractions` generated in response to the original phrase `muscle contractions`), and $\text{max}=31309$ (for the perturbed phrase `know 1` generated in response to the original phrase `know one`). The mean list length is 9259. This variation in the length of our ranked lists practically means that we cannot always use the majority of distance and correlation metrics presented in Section 3.3, because they require equal-sized lists. To address this, we impose a fixed list length of maximum 1000 elements, i.e. we keep, at most, the top-1000 ranked elements in each list. If, when comparing two lists, one contains <1000 elements, we prune both lists to the length of the shortest list. The choice of 1000 as maximum length is an arbitrary choice and is not indicative of any tuning. Fixing the length of our ranked lists to 1000 implies that we represent each phrase by the top 1000 most informative terms found in their context windows.

We compute the distance and correlation between the ranked list of TF-IDF scores of the original phrase and its substitution phrase using all the Class I & II metrics presented in Section 3.3. This outputs a single score per phrase, which we treat as an (inverse for distances or direct for correlations) approximation of the degree of compositionality of that phrase.

5. Experimental evaluation

We evaluate our compositionality detection method on a recent dataset of 1048 2-term phrases (noun-noun) [31]. This is the largest compositionality-annotated dataset we could find. In this dataset, each phrase has four binary compositionality human expert assessments. We report Spearman’s ρ correlation between our method’s decisions on compositionality and the average of the four human annotations of compositionality of that dataset (Table 4).

We report, as state of the art baseline performance, the additive and multiplicative models of Reddy et al. [8] and the best performing deep learning method (sparse interaction) of Yazdani et al. [15]. We do not reimplement these methods; we only report their published scores on the same dataset as per Yazdani et al. [15].

We see in Table 4 that our approach outperforms all baselines using any of the considered metrics. Among these distance and correlation metrics, the Chebyshev and Hausdorff distances score the lowest, but still outperform the baselines. The Chebyshev and Hausdorff distances emphasise the maximal absolute difference in rank and maximal difference in TF-IDF score respectively. It thus appears that this emphasis on maximal differences is not optimal for this setup. One reason could be that we have

Supervised (baselines)	
ADD (Reddy et al., 2011)	0.21
MULT (Reddy et al., 2011)	0.09
Deep Learning (Yazdani et al., 2015)	0.41
Unsupervised (ours)	
Rank or l_1 distance (Equation 1)	0.59
Chebyshev or l_∞ distance (Equation 3)	0.50
CosRank distance (Equation 5)	0.60
Hamming distance (Equation 6)	0.55
Hausdorff distance (Equation 7)	0.50
Pearson correlation coefficient (Equation 8)	0.62
AP correlation coefficient (Equation 9)	0.58

Table 4: Spearman’s ρ correlation between system decisions and human annotations of compositionality (the higher, the better).

trimmed our ranked lists to the top 1000 highest TF-IDF scores, and that among those highest TF-IDF scores, maximal differences may not be as noticeable, as on a much bigger range of TF-IDF scores that represent most levels of term informativeness (as opposed to just the most informative terms).

The best performing metric is the standard Pearson’s correlation, followed closely by the CosRank distance. Pearson’s correlation is the ratio of the covariance of the TF-IDF scores in the two lists over the product of their standard deviation. On a higher level of abstraction, Pearson’s correlation can be seen as analogous to the cosine distance in vector space: the cosine distance measures similarity in vector space, by considering only the non-zero dimensions of (very often) sparse vectors. It is possible to normalise the attribute vectors by subtracting the vector means, in which case one can compute the *centered cosine similarity*, which is equivalent to Pearson’s correlation. Our second best metric, CosRank, is also analogous to the cosine distance as discussed in Section 3.3 and also by Dinu and Ionescu [22]. The fact that our two best performing metrics can be seen as analogous to the cosine distance in vector space indicates that not only the choice of measurement, but also the choice of representation, can greatly affect performance.

Figure 1 plots the human compositionality annotations of the 1042 phrases (y axis, 0 = compositional, 1 = non-compositional) against the distance or correlation of our 7 metrics (x axis). The 1042 data points have been sorted per distance or correlation score and then binned into 11 bins. Each point in Figure 1 corresponds to the mean human compositionality score per bin. The number of bins has been decided using Scott’s formula [32]:

$$M = \frac{R}{3.49s} N^{1/3} \quad (12)$$

where M is the number of bins, R is the range, N is the number of data points, and s is the sample variance. This results in: 10 equal-sized bins of 95 phrases each, and 1 bin of the remaining 92 phrases (this is the bin with the highest distance or correlation).

In Figure 1 ideally we would expect compositionality (x axis) to increase as distance decreases or as correlation increases. We see that this is indeed the case approximately for Rank CosRank, Hamming distance and for Pearson and AP correlation.

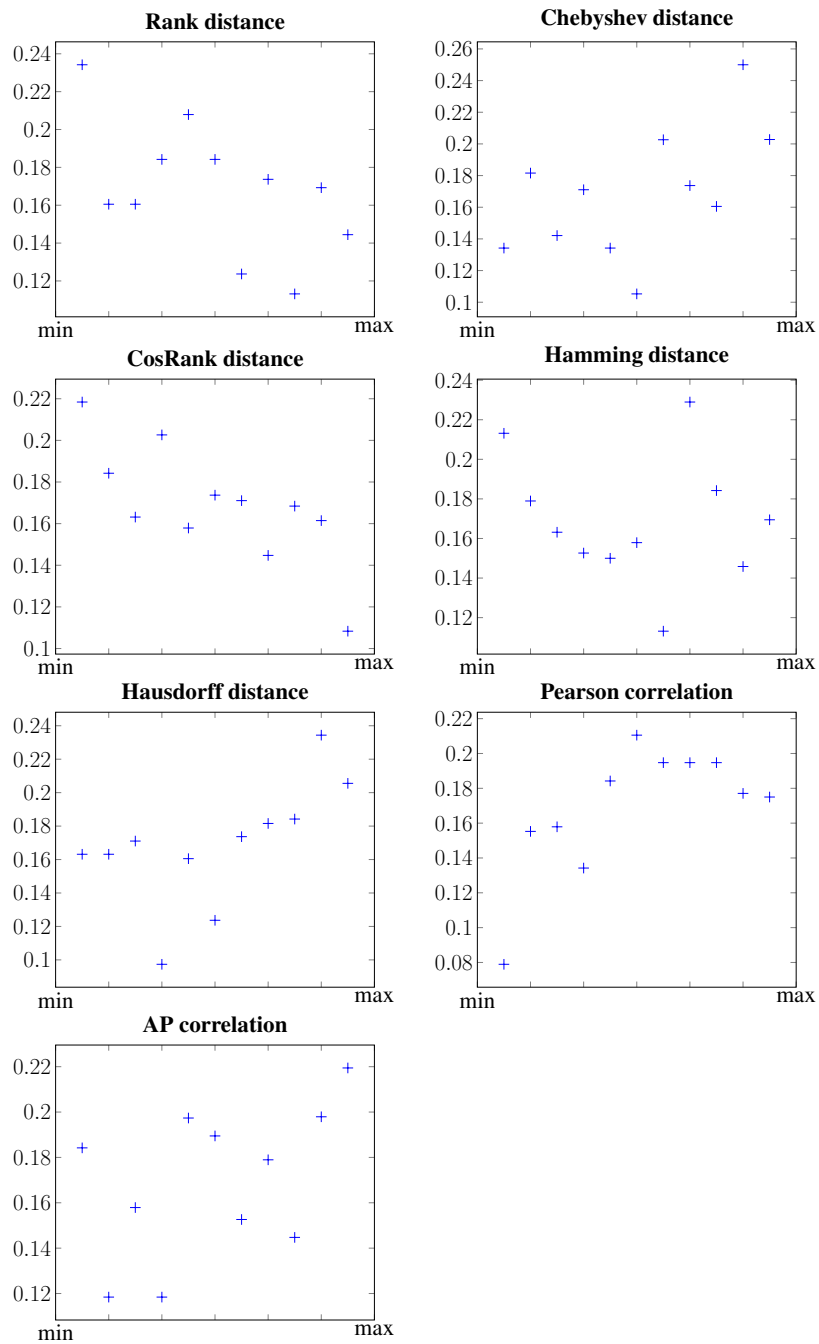


Figure 1: Non-compositionality score approximated by distance or correlation (x axis, binned) versus human compositionality score (y axis).

Non-compositional	
freshman year	highest Minkowski distance
case law	highest Chebyshev distance
body whorl	highest CosRank distance
umbrella organisation	highest Hamming distance
case law	highest Hausdorff distance
vice president	lowest Pearson correlation
chain smoker	lowest AP correlation
Compositional	
mountain goat	lowest Minkowski distance
picnic lunch	lowest Chebyshev distance
goose fossil	lowest CosRank distance
potato peeler	lowest Hamming distance
nightclub goer	lowest Hausdorff distance
school alumni	highest Pearson correlation
flight lessons	highest AP correlation

Table 5: Examples from the top- and bottom-scored phrases by each of our metrics and by humans.

However, for Chebyshev and Hausdorff, we see no such linear trend: the points are generally scattered, and seem to have more of an approximately ascending (as opposed to the expected descending) trend as the y axis increases. This finding agrees with the observation that these two metrics (Chebyshev and Hausdorff) performed the lowest among all our metrics in Table 4. As discussed above, a reason why these two metrics underperform could be their emphasis on maximum score difference between the two lists (which is largely reduced when we trim lists to the top 1000).

Finally Table 5 displays some examples of phrases that were annotated as most or least compositional *both* by humans and also by our metrics (by having the highest/lowest distance/correlation respectively). We see that (a) compositional phrases tend to have more literal meanings than non-compositional, and (b) depending on the degree of (non-)compositionality, its detection may be a hard task even for humans (e.g. case law, goose fossil).

6. Discussion of limitations

As all substitution-based methods for compositionality detection, our method can also be criticised for being unable to discriminate non-compositional phrases from collocational phrases, because they both share the same property of non-substitutability (their constituents cannot be replaced with their synonyms) [15]. This criticism is related to the venerable *principle of semantic substitutivity*, first formulated by Husserl (1913) [33]: two phrases belong to the same semantic category if they are intersubstitutable within any meaningful expression *salva significatione*. This principle is considered controversial, because there are many synonyms that are not everywhere intersubstitutable [33]. To our knowledge, this remains an open problem in substitution-based compositionality detection.

Our method estimates the degree of compositionality of isolated phrases, following the current experimental practice of using mainly 2-term phrases. Applying the same

method to phrases that are embedded into fully formed sentences may be problematic. For instance, problems may arise when the phrase whose compositionality we detect is a constituent of a larger expression. Borrowing an example from [34], measuring the semantic similarity of:

Plato was bald

with

baldness was an attribute of Plato

could lead to misleading inferences about the semantic similarity of:

the philosopher whose most eminent pupil was Plato
was bald

and

the philosopher whose most eminent pupil was baldness
was an attribute of Plato.

In this case, the second sentence, not only has a different meaning than the first sentence, but also is semantically non-sensical. It remains to be investigated to what extent and under which conditions phrase-level compositionality estimation can be applied to full sentences. The absence of large human-annotated bespoke datasets for this task remains a problem for such investigations.

7. Conclusion

We presented a method for estimating degrees of compositionality in phrases. Our method is based on the premise that substitution by synonyms is meaning-preserving [3], and estimates compositionality as the semantic similarity between a phrase and a version of that phrase where words have been replaced by their synonyms [1]. Unlike previous approaches that represent such phrases (original and substitution-formed) as vectors or language models, we represent them as ranked lists. This ranked list representation is a novel contribution. The elements of these lists are contextual terms extracted from some appropriate corpus and ranked according to their TF-IDF (any other term informativeness score can be used). Moving to ranked list representations allows us to approximate the semantic similarity between two phrases using a range of well-known and, we argue, more refined distance and correlation metrics, designed specifically for lists. We review a number of these metrics and experimentally show that they outperform state-of-art baselines for this task.

Acknowledgements

Work partially supported by C. Lioma’s FREJA research excellence fellowship (grant no. 790095). We thank Jakob Grue Simonsen for thoughtful comments and valuable insights.

References

References

- [1] D. Kiela, S. Clark, Detecting compositionality of multi-word expressions using nearest neighbours in vector space models, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Seattle, Washington, USA, 2013, pp. 1427–1432.
- [2] C. Lioma, J. G. Simonsen, B. Larsen, N. D. Hansen, Non-compositional term dependence for information retrieval, in: R. A. Baeza-Yates, M. Lalmas, A. Moffat, B. A. Ribeiro-Neto (Eds.), Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015, ACM, 2015, pp. 595–604.
- [3] D. Lin, Automatic identification of non-compositional phrases, in: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, College Park, Maryland, USA, 1999, pp. 317–324.
- [4] T. Baldwin, C. Bannard, T. Tanaka, D. Widdows, An empirical model of multi-word expression decomposability, in: Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment, Association for Computational Linguistics, Sapporo, Japan, 2003, pp. 89–96.
- [5] K. Sparck-Jones, A statistical interpretation of term specificity and its application in retrieval, *Journal of Documentation* 28 (1) (1972) 11–21.
- [6] G. Katz, E. Giesbrecht, Automatic identification of non-compositional multi-word expressions using latent semantic analysis, in: Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties, Association for Computational Linguistics, Sydney, Australia, 2006, pp. 12–19.
- [7] S. Venkatapathy, A. Joshi, Measuring the relative compositionality of verb-noun (v-n) collocations by integrating features, in: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Vancouver, British Columbia, Canada, 2005, pp. 899–906.
- [8] S. Reddy, D. McCarthy, S. Manandhar, An empirical study on compositionality in compound nouns, in: Proceedings of 5th International Joint Conference on Natural Language Processing, Asian Federation of Natural Language Processing, Chiang Mai, Thailand, 2011, pp. 210–218.
- [9] D. McCarthy, S. Venkatapathy, A. Joshi, Detecting compositionality of verb-object combinations using selectional preferences, in: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 369–379.

- [10] P. Cook, A. Fazly, S. Stevenson, Pulling their weight: Exploiting syntactic forms for the automatic identification of idiomatic expressions in context, in: *Proceedings of the Workshop on A Broader Perspective on Multiword Expressions*, Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 41–48.
- [11] R. Socher, B. Huval, C. D. Manning, A. Y. Ng, Semantic compositionality through recursive matrix-vector spaces, in: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Association for Computational Linguistics, Jeju Island, Korea, 2012, pp. 1201–1211.
- [12] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: C. J. C. Burges, L. Bottou, Z. Ghahramani, K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, 2013, pp. 3111–3119.
- [13] B. Salehi, P. Cook, T. Baldwin, A word embedding approach to predicting the compositionality of multiword expressions, in: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Denver, Colorado, 2015, pp. 977–983.
- [14] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, *CoRR* abs/1301.3781.
- [15] M. Yazdani, M. Farahmand, J. Henderson, Learning semantic composition to detect non-compositionality of multiword expressions, in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 1733–1742.
- [16] R. Kumar, S. Vassilvitskii, Generalized distances between rankings, in: Rappa et al. [35], pp. 571–580.
- [17] R. Fagin, R. Kumar, D. Sivakumar, Comparing top k lists, *SIAM J. Discrete Math.* 17 (1) (2003) 134–160.
- [18] S. Vigna, A weighted correlation index for rankings with ties, in: A. Gangemi, S. Leonardi, A. Panconesi (Eds.), *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, ACM, 2015, pp. 1166–1176.
- [19] M. Deza, T. Huang, Metrics on permutations, a survey, *J. Combin., Inf. Syst. Sci.* 23 (1998) 173–185.
- [20] M. Sun, G. Lebanon, K. Collins-Thompson, Visualizing differences in web search algorithms using the expected weighted hoeffding distance, in: Rappa et al. [35], pp. 931–940.

- [21] A. Ciobanu, A. Dinu, Alternative measures of word relatedness in distributional semantics, in: *Proceedings of the Joint Symposium on Semantic Processing. Textual Inference and Structures in Corpora*, 2013, pp. 80–84.
- [22] L. P. Dinu, R. Ionescu, Clustering methods based on closest string via rank distance, in: A. Voronkov, V. Negru, T. Ida, T. Jebelean, D. Petcu, S. M. Watt, D. Zaharie (Eds.), *14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2012*, Timisoara, Romania, September 26–29, 2012, IEEE Computer Society, 2012, pp. 207–213.
- [23] O. Etesami, A. Gohari, Maximal rank correlation, *IEEE Communications Letters* 20 (1) (2016) 117–120.
- [24] E. Yilmaz, J. A. Aslam, S. Robertson, A new rank correlation coefficient for information retrieval, in: S. Myaeng, D. W. Oard, F. Sebastiani, T. Chua, M. Leong (Eds.), *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008*, Singapore, July 20–24, 2008, ACM, 2008, pp. 587–594.
- [25] C. Lioma, R. Blanco, Part of speech based term weighting for information retrieval, in: M. Boughanem, C. Berrut, J. Mothe, C. Soulé-Dupuy (Eds.), *Advances in Information Retrieval, 31th European Conference on IR Research, ECIR 2009*, Toulouse, France, April 6–9, 2009. *Proceedings*, Vol. 5478 of *Lecture Notes in Computer Science*, Springer, 2009, pp. 412–423. doi:10.1007/978-3-642-00958-7_37. URL http://dx.doi.org/10.1007/978-3-642-00958-7_37
- [26] C. Lioma, I. Ounis, Extending weighting models with a term quality measure, in: N. Ziviani, R. A. Baeza-Yates (Eds.), *String Processing and Information Retrieval, 14th International Symposium, SPIRE 2007*, Santiago, Chile, October 29–31, 2007, *Proceedings*, Vol. 4726 of *Lecture Notes in Computer Science*, Springer, 2007, pp. 205–216. doi:10.1007/978-3-540-75530-2_19. URL http://dx.doi.org/10.1007/978-3-540-75530-2_19
- [27] C. Lioma, C. J. K. van Rijsbergen, Part of speech n-grams and information retrieval, *Revue française de linguistique appliquée XIII* (1) (2008) 9–11.
- [28] R. Blanco, C. Lioma, Random walk term weighting for information retrieval, in: W. Kraaij, A. P. de Vries, C. L. A. Clarke, N. Fuhr, N. Kando (Eds.), *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Amsterdam, The Netherlands, July 23–27, 2007, ACM, 2007, pp. 829–830. doi:10.1145/1277741.1277930. URL <http://doi.acm.org/10.1145/1277741.1277930>
- [29] R. Blanco, C. Lioma, Graph-based term weighting for information retrieval, *Inf. Retr.* 15 (1) (2012) 54–92. doi:10.1007/s10791-011-9172-x. URL <http://dx.doi.org/10.1007/s10791-011-9172-x>

- [30] C. Lioma, B. Larsen, W. Lu, Rhetorical relations for information retrieval, in: W. R. Hersh, J. Callan, Y. Maarek, M. Sanderson (Eds.), The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012, ACM, 2012, pp. 931–940. doi:10.1145/2348283.2348407. URL <http://doi.acm.org/10.1145/2348283.2348407>
- [31] M. Farahmand, A. Smith, J. Nivre, A multiword expression data set: Annotating non-compositionality and conventionalization for english noun compounds, in: Proceedings of the 11th Workshop on Multiword Expressions, Association for Computational Linguistics, Denver, Colorado, 2015, pp. 29–33.
- [32] D. W. Scott, On optimal and data-based histograms, *Biometrika* 66 (1979) 605–610.
- [33] Z. G. Szabó, Compositionality, in: E. N. Zalta (Ed.), The Stanford Encyclopedia of Philosophy, fall 2013 Edition, 2013.
- [34] P. Geach, Logical procedures and the identity of expressions (1965) 108–115 Reprinted in *Logic Matters*, Berkeley, CA: University of California Press, 1980.
- [35] M. Rappa, P. Jones, J. Freire, S. Chakrabarti (Eds.), Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010, ACM, 2010.